2008 International Conference on Information Processing in Sensor Networks

Towards Energy Efficient Design of Multi-Radio Platforms for Wireless Sensor Networks

Dimitrios Lymberopoulos
Yale University
New Haven, CT, 06511
dimitrios.lymberopoulos@yale.edu

Nissanka B. Priyantha, Michel Goraczko and Feng Zhao
Microsoft Research
Redmond, WA, 98052
{bodhip,michelg,zhao}@microsoft.com

Abstract

We study the problem of concurrently supporting multiple radios with different capabilities and interfaces on a single sensor node platform. Through a detailed experimental study on hardware multi-radio platforms, using the two representative radio technologies 802.15.4 and 802.11, we identify bottlenecks and design tradeoffs that are usually overlooked and that, as we show, have a significant impact on the sensor network's performance and energy efficiency. Our findings are threefold. We show that a proper pairing of processor and radio is crucial for taking the full advantage of the energy efficiency of higher bandwidth radios. The processor/radio pairing affects the energy balance of a sensor node, thus making the design of dynamic switching among multiple radios more challenging. Second, we demonstrate and quantify the impact of network traffic on energy consumption of a sensor node while varying network parameters, and illustrate the deficiency of existing energy-optimizing protocols. Our results indicate that by properly adjusting network parameters, such as packet size and transmission period, energy savings of up to 50% can be achieved under heavy network traffic conditions when a CSMA-based MAC is used. We conclude by presenting a set of guidelines for designing and implementing energy efficient multi-radio platforms.

1 Introduction

The adoption of battery operated sensor networks for real world applications depends heavily on the battery life of sensor nodes. Given the slow advance in battery technology (battery capacity has only been doubled over the last 10 years [17]) and the ever increasing complexity of sensing, processing and communication components, efficient energy management becomes an essential component of every sensor platform. Many research efforts have focused on multiple aspects of energy management, such as low power sensor node design, energy harvesting, and efficient scheduling at the node and network level. Most of these efforts focus on reducing the energy spent on wireless communication since this is the most energy demanding component on a typical sensor node.

The use of heterogeneous radios on a single sensor platform becomes an increasingly popular mechanism for reducing wireless communication energy dissipation [6, 8, 16, 18, 5]. The main idea behind this mechanism is to opportunistically use two (or more) radios with different energy and throughput characteristics to minimize the total energy consumption. For instance, Chipcon's 802.15.4 compliant radio is a low power radio that can provide sufficient data rates (250Kbps) for many sensor network applications. However, it is less energy efficient compared to high-bandwidth radios such as 802.11b. The 802.11b radios have higher power consumption but they provide significantly higher data rates (11Mbps). This means that high-bandwidth radios can transmit more data in less amount of time compared to low power radios and they can be more energy efficient. For example, the energy per bit of the 802.15.4 radio (979nJ/bit) is almost 9 times higher than the energy per bit of an 802.11b radio (112nJ/bit)[18].

Even though the energy per bit metric provides a baseline for comparing the energy efficiency of different radios, it fails to capture various system aspects that affect radio's energy dissipation. Besides high power consumption, radios like 802.11b have a large startup time. In other words, the time it takes to power up and configure the radio is orders of magnitude higher (approximately 2 to 3s) compared to that of low power radios like 802.15.4 (less than 2ms). This creates a fixed energy overhead that takes place every time we power up the radio and that is independent of the size of data to be transmitted. Under the presence of this high startup cost, high bandwidth radios become more energy efficient only when a large number of bytes have to be transmitted. In that way, the high startup cost gets amortized as more and more bytes are transmitted. The exact number of bytes above which 802.11b radio becomes more energy efficient is usually called *the break-even point be-*

978-0-7695-3157-1/08 $25.00 © 2008 IEEE
DOI 10.1109/IPSN.2008.51

tween the two radios. In the case of 802.11b and 802.15.4 radios, the break-even point is usually large due to the huge startup cost of the 802.11b radio.

Assuming the application data rate can be supported by either of the two radios, the conventional approaches to selecting one radio over the other are primarily based on the per bit and startup energy consumption of the two radios [18, 12]. However, based on experimental data acquired on hardware we identify that there are other factors such as the sensor node hardware architecture, the background network traffic, and the network topology that significantly affect the energy consumption of a given radio. Using a heterogeneous sensor node equipped with an 802.15.4 and an 802.11b radio, we performed an extensive set of experiments, while varying various network parameters, that revealed several important system design issues and guidelines for the energy efficient use of multiple radios. Our experimental findings provide valuable insight on the following key design issues:

(1) How should multiple heterogeneous radios be interfaced to one or more processors on a single sensor node in order to achieve maximum energy efficiency? In general, most simplified energy dissipation models assume that we can transparently swap the low and high power radios on a low end sensor node and still take full advantage of both radios. We show that radios spend a significant amount of time interacting with the processor. The process of configuring and writing packets to the radio can be time and therefore energy consuming depending on the specific interface of the radio and the speed of the processor to which the radio is interfaced. A mismatch between the processor's capabilities and the complexity of radio's interface can have a significant impact on radio's energy consumption.

(2) How is the break-even point affected by the way the two radios are interfaced on the sensor node? In other words, what parameters should the sensor node take into account so that it knows when it is energy efficient to use the high power radio? Through detailed measurements and extrapolated data, we identify the major design bottlenecks and tradeoffs that affect the break-even point between two of the representative radios, the 802.11b and 802.15.4, and that apply to any typical multi-radio platform similar to the one used in our experiments.

(3) What are the major system-level bottlenecks and parameters that we have to model in order to get accurate energy analysis of multi-radio platforms? Conventional radio energy consumption models focus mostly on the energy consumption during active RF transmission or reception, and on the energy consumption for powering up the radio. However, given that specific radios have to be matched with specific processors in order to maximize their energy efficiency, the energy overhead of the matching processor has to be also taken into account when considering radio's energy overhead. Furthermore, we quantify the effect of background network traffic on radio's energy consumption for different network parameters, and we show that the radio packet size has a substantial impact on radio's energy consumption. In particular, we demonstrate that larger packet sizes result in significant energy consumption due to repeated transmission attempts. We suggest a measurement-based technique at the MAC layer for selecting the appropriate packet size under given network conditions at runtime that as we show can lead to energy savings of up to 50% when a CSMA-based MAC is used. We also discuss the effect of network topology and radio transmission range on radio's energy dissipation.

Before delving into the details of our work we have to first clarify its scope. First, our goal is not to just compare two radios with different characteristics. We study the different design bottlenecks and tradeoffs that will allow us to concurrently use both radios to maximize the energy efficiency of a multi-radio sensor node platform. Second, while the absolute numbers presented in this study are specific to the hardware used, the tradeoffs and bottlenecks revealed and analyzed in this paper should hold for any similar multi-radio platform. Third, our focus is not on providing optimal solutions to the dynamic radio switching problem at the application level that would take into consideration the operating system, middleware and application stack properties. Rather, it is on the underlying hardware configuration bottlenecks and tradeoffs.

The rest of the paper is organized as follows. Section 2 provides an overview of the related work. Sections 3 and 4 describe the hardware and software infrastructure used in the experiments presented in this paper. Section 5 examines how the processor speed and the hardware interface between the radio and the processor affects the energy consumption of the 802.15.4 and 802.11b radios . In Section 6 we present a detailed characterization of the impact of the background network traffic on the energy consumption of a radio. Section 7 describes various parameters that should be considered when selecting a radio from multiple available radios. Section 8 concludes the paper by reviewing the main findings of our work and discussing their implications on the design and energy management of wireless sensor networks.

2 Related Work

Several research efforts have focused on using multiple radios to implement "wake-on-wireless", where a low power radio is used to wake up a high power radio [15, 13]. In contrast to efficiently waking up a high power radio, our work examines how to efficiently select among multiple radios and radio parameters for minimizing data transmission's energy consumption. To the best of our knowledge, this is the first work that utilizes an actual multi-radio sensor node to study the problem of energy-efficient radio selection.

Other research projects in wireless sensor networks use a combination of high power and low power radios where the high power radios implement a high speed backbone for efficient data transfer. Wan *et al.* use a collection of line-powered nodes that contain both 802.11 and 802.15.4 radios that acts as virtual sinks to reduce network congestion, where the 802.11 radios implement a high speed backbone [19]. Stathopoulos *et al.* use a network of sensor nodes with both 802.15.4 radios and 802.11 radios. The 802.15.4 radios implement a constantly available multi-hop network while 802.11 radios are woken up to implement an "on demand" end to end communication path with high data rates [18]. In our work, we treat both 802.11 and 802.15.4 radios as equal and examine the radio and parameter selection for achieving energy efficient data transmission.

The work by Pering *et al.* is perhaps the closest to our work in terms of selecting an appropriate radio for minimizing energy [12]. The authors suggest that, when selecting a radio for communication, the low power Bluetooth radio must be used as long as the available bandwidth can support application needs rather than examining the most efficient radio selection under given conditions. Our work shows that these design decisions are not always the most energy efficient, and examines how to select the most energy efficient radio based on the sensor node architecture and the runtime network parameters.

More complementary rather than comparable to our work, are the research efforts in quantifying the energy overhead caused when large packets are used due to frequent bit errors [4, 11]. This type of analysis affects the design of the MAC layer. We focus on the effect of application level parameters such as the packet size and the packet transmission period on the energy consumption of CSMA protocols due to multiple transmission attempts.

Some solutions aimed at reducing radio energy propose new MAC protocols that adjust their parameters appropriately to reduce multiple retransmission attempts [3]. Our approach is to modify application level parameters, such as packet payload size, while being compatible with 802.15.4 and 802.11 MAC layer specifications.

3 Hardware Infrastructure

The centerpiece of our infrastructure is the mPlatform sensor node [6]. The mPlatform consists of a collection of stackable hardware modules with a common hardware interface. Some of the mPlatform modules are general purpose processing boards while others are special purpose modules such as radio boards, sensor boards, and power boards. A high speed parallel bus enables fast and scalable communication between different modules. The mPlatform hardware architecture enables us to create custom sensor node configurations with heterogeneous processing and/or communication capabilities by stacking together the appropriate modules. For our experiments, we used an mPlatform node equipped with an 802.15.4 radio module, an 802.11b radio module, and a power module (Figure 1).

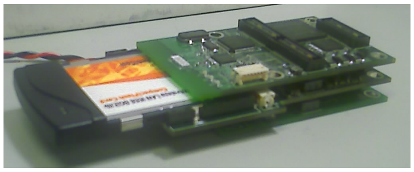

Figure 1. An mPlatform node with the 802.11 and 802.15.4 radios.

The 802.15.4 radio module uses the CC2420 radio chip from Chipcon. This radio chip is interfaced to a low power 16-bit microcontroller, the MSP430F1611 from Texas Instruments, located on the same module. The microcontroller operates at 6MHz and is interfaced to the radio chip using the SPI bus and several GPIO pins. On the microcontroller side, we have implemented our own MAC layer that is *compliant with the CSMA portion of the 802.15.4 specification*. The use of an 802.15.4 compliant MAC makes our node interoperable with other widely used sensor nodes such as Telos, MicaZ and XYZ. Hence our experimental results are directly applicable to a variety of current state-of-the art sensor platforms. The CC2420 radio has a maximum effective data rate of 250Kbps, a maximum packet size of 128 bytes, and low current consumption that ranges 10–18mA for transmitting data and up to 20mA for listening/receiving data.

The 802.11b radio module consists of a WL1100C-CF radio card from Ambicom [2]. It supports a data rate of 11Mbps and a maximum packet size of 1500 bytes while consuming approximately 350mA for transmitting and 230mA for receiving data. The card is interfaced to an MSP430F1611 microprocessor that uses its GPIO pins to control the 802.11b radio by driving the appropriate pins on the CompactFlash interface of the radio card. In the case of the 802.11 radio the actual MAC layer is already implemented on the baseband processor of the 802.11 card. As a result of this, there is no software MAC layer running on the microprocessor. Note, that the absolute measurements presented in this paper will only apply to the specific card since other 802.11-based cards might have slightly different characteristics. However, the broader design bottlenecks and tradeoffs demonstrated in this paper suggest several system design guidelines that should be exercised when interfacing radios to processors.

4 Experimental Setup

In all of our experiments, we opted not to use an operating system on our sensor node. Instead, a simple C program containing the drivers for each radio and the MAC layer for

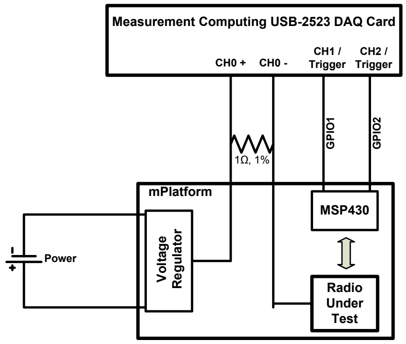

Figure 2. The setup used for measuring power consumption and timing of radio events.

the 802.15.4 radio was used. This allowed us to acquire accurate timing and power information for both radios ignoring possible operating system overheads and limitations.

All the measurements presented in this paper were acquired using the MSP430F1611 microprocessor. Whenever data is provided or displayed for another processor (e.g. ARM7TDMI or PXA) to demonstrate design bottlenecks and tradeoffs, this data was extrapolated (and not measured) using the actual measured data of the MSP430 processor.

The hardware configuration shown in Figure 1 was used for all of the experiments described in Sections 5 and 6. During each experiment only one of the two radio modules was used.

Time and power information for each radio chip was recorded as shown in Figure 2. To obtain power consumption, we acquired the current consumption of each radio by measuring the voltage drop across a 1 Ohm (1%) resistor connected in series with the power supply of each radio. We measured the voltage drop using a high-speed data acquisition (DAQ) card [9]. Two GPIO pins of the microprocessor controlling the radio were used to signal the occurrence of different events of interest (e.g. radio packet was submitted to the MAC layer, radio packet was sent, radio packet was dropped etc.). The DAQ card was continuously recording all input signals at a rate of 75000 or 25000 samples per second depending on the type of the experiment. After each experiment, we were able to reconstruct the exact timing of each event and compute its power consumption by analyzing the firing patterns of the GPIO pins and the measured current consumption. Given that the packets that are submitted to the MAC layer are served in a FIFO order we can easily link these events over time and assign them to specific packets. In addition, since we were recording all input signals concurrently and at a constant high rate, we were able to easily estimate the time that elapsed between events using the timestamps provided by the DAQ card.

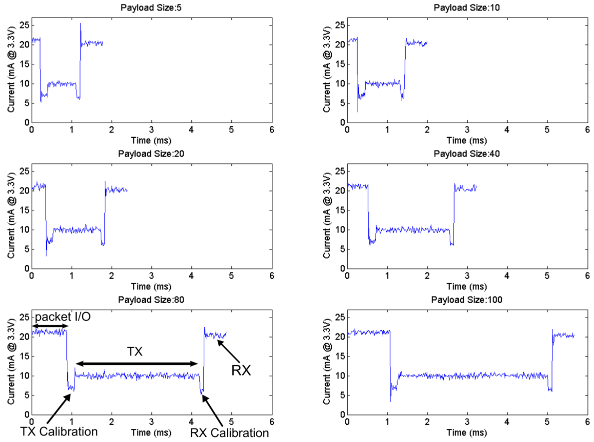

Figure 3. The 802.15.4 current vs. payload.

This setup enabled us to accurately compute the transmission time of every radio packet submitted to the MAC layer. In the rest of the paper we use the term *transmission time of a packet* to refer to *the difference between the time that the processor starts sending a packet to the MAC layer and the time the last bit of that packet appears on the radio channel*. Note that this time interval includes the time it takes the processor to construct the packet (e.g. create headers etc.), write the packet to the radio, and the time to transmit the packet over the radio channel.

In the case of Chipcon's CC2420 radio we used 802.15.4 compliant packets including an 11-byte header, followed by the packet payload, and the Frame Control Sequence (FCS) and CRC bytes. To do a fair comparison between the two radios, we used plain Ethernet packets for the 802.11b radio. This eliminated the need for sending large TCP/IP headers that can significantly increase the size of each packet. Every 802.11 packet contains the destination and source Ethernet addresses (7 bytes each), the protocol type (2 bytes), the payload length (2 bytes), and the payload.

5 Radio Energy Profiling

The goal of our first set of experiments was to characterize and compare the performance of the two radios under no background network traffic. We interfaced an mPlatform node to the DAQ card as shown in Figure 2. Then using each radio, we transmitted packets while varying their payload sizes. For each packet size, 10 different packets were transmitted. Figure 3 shows the current consumption of the 802.15.4 radio for different payload sizes at the lowest transmission power level; Figure 4 shows the current consumption of the 802.11 radio for different payload sizes.

For the 802.15.4 radio, the different states of the packet transmission process are visible in Figure 3. First, the packet is constructed and written to the radio over the SPI interface. During this phase the receive circuitry is turned on resulting in a current consumption of $\sim$20mA. Once the packet has been written, the send command is sent, resulting in the transmission of the RF packet. The current consumption during packet transmission is $\sim$ 10mA because of the lowest transmit power level we used to easily identify the different phases of the packet transmission process. The highest transmit power level results in a current consumption of ∼18mA. At the end of the RF transmission, the radio returns to its default state of listening to the channel. We observe that *two* time intervals increase with the payload size: the time interval for sending the packet to the radio, and the time interval for completing the RF transmission. Note that constructing and writing the packet to the radio corresponds to ∼22% of the total time taken for completing the actual RF packet transmission. This shows that simply considering the RF transmission time of a packet gives a very optimistic value for the radio energy consumption [1].

In Figure 4 we observe a similar effect for the 802.11 radio. However, now the actual transmission time of the packet is extremely small compared to the time it takes for the processor to write the packet into the radio. Since the 802.11b radio transmits at a data rate of 11Mbps, the actual transmission time of a specific number of bytes is very small compared to that of the 802.15.4 radio which can only offer a maximum data rate of 250Kbps. Thus, the ratio of the time to write the packet over the time to actually transmit the packet is much higher in the case of the 802.11b radio. In practice, this results in prolonging the time that the high bandwidth radio is operating in idle mode. Given the extremely high current consumption of 802.11b in idle mode ($\sim 235mA$), *it is apparent that ignoring the energy dissipation due to writing the packet into the radio results in a completely inaccurate estimation of the operational cost for the 802.11b radio.*

An even more interesting conclusion can be drawn by comparing the time it takes to write a packet of the same size for both radios. Figures 3 and 4 clearly show that the time it takes to write a packet of the same size to the 802.11b radio can be up to 4 times higher than the time it takes for the 802.15.4 radio. This is due to the fact that the two radios have completely different I/O interfaces. The 802.15.4 radio provides a very simple and efficient interface that was specifically designed for low-end microprocessors. On the other hand the interface of the 802.11b radio is much more complex including the need for allocating/deallocating buffer space on the baseband processor before the microcontroller can actually start writing data to it. This, along with the fact that the MSP430F1611 microprocessor, used to control the radio, operates at 6MHz, creates a major bottleneck on the node's communication path. As we will show in the next section, this inefficiency in the I/O interface of the 802.11 radio and the MSP430 processor, combined with its high power consumption (approximately 14 times higher than that of the 802.15.4) can completely eliminate the energy efficiency advantage of a high bandwidth radio.

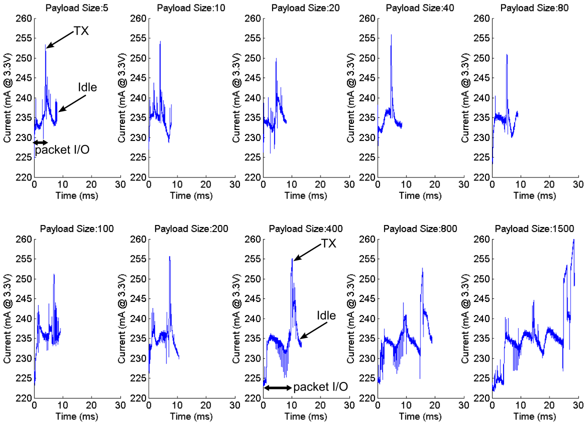

Figure 4. The 802.11 current vs. payload.

5.1 Ideal Break-Even Point Calculation

By fitting a linear curve to the measurements shown in Figures 3 and 4 we can compute the transmission time (T) and the energy consumption (E) when sending a single packet as a function of packet's payload size for both radios:

$$T_{802.15.4}(ms) = 0.041 \times x + 0.91, 5 \leq x \leq 117 \quad (1)$$

$$T_{802.11}(ms) = 0.014 \times x + 3.8, 5 \leq x \leq 1500 \quad (2)$$

$$E_{802.15.4}(mJ) = 0.0026 \times x + 0.05, 5 \leq x \leq 117 \quad (3)$$

$$E_{802.11}(mJ) = 0.011 \times x + 3, 5 \leq x \leq 1500 \quad (4)$$

A graphical representation of these equations is shown in Figures 5 and 6. Using equations (1), (2), (3), and (4) we can calculate the time and energy required by the two radios when transmitting an arbitrary number of bytes. Figures 7 and 8 show the results of these calculations assuming that maximum size packets are used for both radios (128 bytes for 802.15.4 and 1500 bytes for 802.11). As Figure 7 shows, when the total numbers of bytes to be sent increases above the maximum number of bytes that can be sent in a single 802.15.4 packet, the transmission time of the 802.11 radio becomes less than the transmission time of the 802.15.4 radio. Furthermore, this transmission time improvement increases as the total number of bytes sent increases. In particular, when the number of bytes to be sent is equal to 16KB we get a performance improvement of a factor of 4. This is due to the fact that the 802.11 maximum packet size is more than 11 times larger than that of the 802.15.4 (1500 bytes vs. 128 bytes). Therefore, the overhead of structuring the packet and writing/sending the packet header is incurred less number of times with the 802.11 radio.

Even though we get an improvement in terms of transmission time when using the 802.11 radio, the radio energy dissipation does not follow the same trend. As Figure 8

---

[1] This is true even though it is possible that the packet construction and packet RF transmission overlap under very specific communication patterns reducing the packet construction overhead.

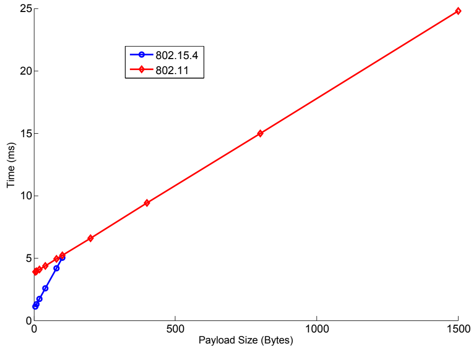

Figure 5. Single packet transmission time vs. payload size.

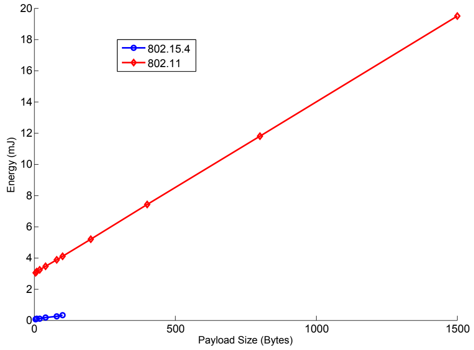

Figure 6. Single packet energy consumption vs. payload size.

shows, the use of the 802.11 radio does not reduce the energy consumption. Note that the 802.15.4 radio is always more energy efficient than the 802.11 radio (when 802.11 is interfaced to the MSP430 processor), and that the performance gap between the two radios increases as the number of bytes to be sent increases. *This means that there is no break-even point between the two radios!* This is caused by two important factors:

1. The operating power consumption of the 802.11 is significantly higher than the power consumption of the 802.15.4. As a result of this, the power inefficiency of the 802.11b radio dominates its transmission time efficiency. For instance, according to Figure 7, the data transmission time for sending 16KB of data using 802.11 is 4 times lower than the data transmission time of the 802.15.4. However, as Figures 3 and 4 indicate, the 802.11b can be up to 11 to 12 times more power hungry than the 802.15.4. As a result of this: $\frac{E_{802.11b}}{E_{802.15.4}} = \frac{P_{802.11b}*T_{802.11b}}{P_{802.15.4}*T_{802.15.4}} \approx \frac{12*P_{802.15.4}*\frac{T_{802.15.4}}{4}}{P_{802.15.4}*T_{802.15.4}} = 3 > 1$.

2. Due to the complex I/O interface of the 802.11 radio and the slow microprocessor used, the time taken for writing a packet to the radio is much longer than the corresponding time taken for the 802.15.4 radio. This, combined with the high power consumption of the 802.11 leads to a significant energy consumption overhead compared to the 802.15.4 radio.

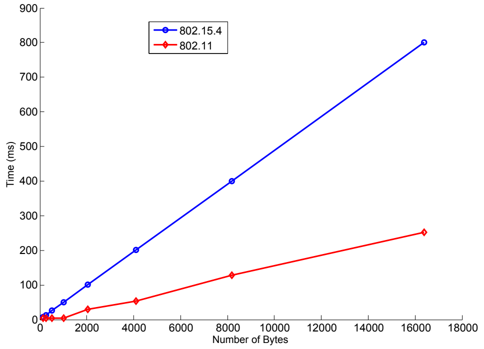

Figure 7. Projected data transmission times vs. data size.

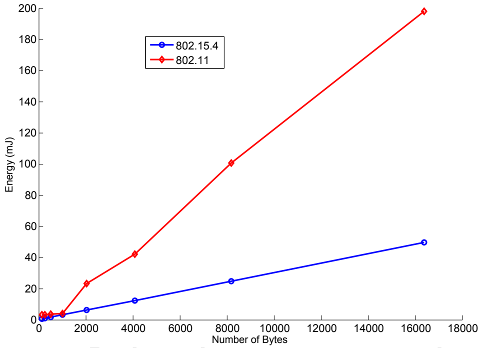

Figure 8. Projected energy consumption vs. data size.

Consequently, the combination of (1) the high power consumption of 802.11 and (2) the increased packet I/O time due to the inefficient I/O interface of 802.11 and the low-end microprocessor controlling the 802.11, makes 802.11 always less energy efficient than 802.15.4.

5.2 Sensor Node Design and Modeling Implications

The experimental results in the previous section demonstrated that 802.11b is less energy efficient than 802.15.4. This is mainly due to the use of a low end processor to control the 802.11b radio. Such a processor is incapable of efficiently managing radio's I/O interface, thus resulting in a significant increase in the packet I/O time. During this time, the radio has to be in its idle state drawing more than 230mA from the power supply.

This observation highlights that not every processor is capable of efficiently driving a given radio. Instead, a proper matching between a heterogeneous set of processors and radios is required. In order to take advantage of the low per-bit energy of high bandwidth radios, higher-end processors operating at sufficiently high speeds have to be

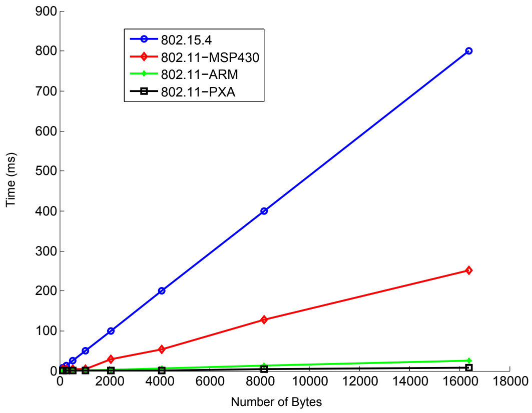

Figure 9. Projected transmission time vs. number of bytes for different processors and radios when maximum packet sizes are used.

used. To better visualize this consider Figure 9 where the overall transmission time for different data sizes is shown when different processors are used. The results for 3 widely used processors are shown: (1) the MSP430F1611 running at 6MHz [6, 14], (2) the ARM7TDMI running at 60MHz [7] and (3) the Intel PXA processor running at 200MHz [1]. For the ARM7TDMI and PXA processors, the data transmission times were extrapolated from the measurements taken with the MSP430F1611 processor. It is clear, that when a faster processor is used, the overall data transmission time is dramatically reduced.

As Figure 10 shows this has a tremendous impact on the energy consumption of the 802.11b radio. The fact that we can minimize the idle time of the radio results in a large energy reduction. The faster the processor we use, the smaller the idle time and therefore the smaller the radio energy consumption.

Even though Figure 10 compares the energy consumption of the two radios under different configurations, it ignores two important parameters: (1) the startup cost of the two radios and (2) the significantly different transmission ranges of the two radios. On one hand, the 802.11b has a large startup energy overhead compared to the 802.15.4 radio. On the other hand, the transmission range of the 802.11 radio is typically 3 times larger than the transmission range of the 802.15.4 radio. Therefore, when considering a network of nodes, in the worst case, 802.15.4 might require 3 times as many sensor nodes (and thus 3 times as many transmissions) as 802.11.

Figure 11 shows a comparison between the energy consumption for the two radios while taking into account both the startup energy as well as the RF range. Note that now we can clearly identify the break-even points for the two radios under multiple configurations. Initially, due to the high startup energy overhead, 802.11b is less energy efficient. However, as the amount of data to be transmitted increases, this cost gets amortized and eventually 802.11b becomes more energy efficient than the 802.15.4 radio. Note that this break-even point is reached at a smaller number of bytes when 802.11b is interfaced to a faster processor or when the smaller transmission range of the 802.15.4 is taken into account. For instance, when the 802.11b is interfaced to the ARM processor, it becomes more energy efficient than the 802.15.4 only when more than 1.15 million bytes have to be transmitted. When the ranging effect is taken into account this number drops down to 250000 bytes. When the PXA processor is used to control the 802.11b radio, these break-even points become even smaller: 745000 and 230000 bytes respectively (we note that these numbers still assume a GPIO driven CF interface implementation by the ARM and PXA processors; a comparable processor with a special purpose hardware CF interface will reach the break-even point at a smaller number of bytes).

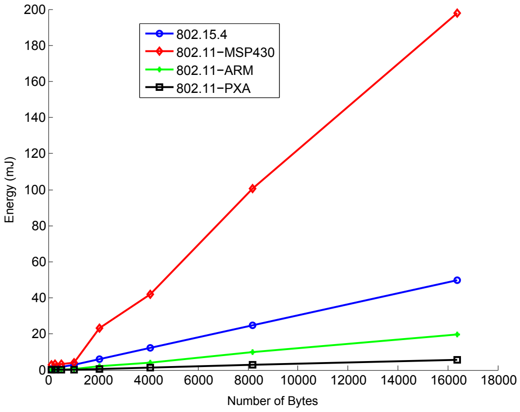

Figure 10. Projected energy vs. number of bytes for different processors and radios when maximum packet sizes are used.

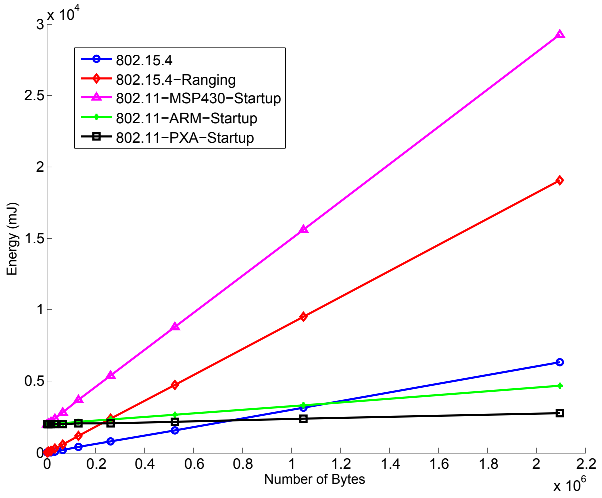

Figure 11. Total energy consumption for both radios under different configurations.

However, note that even when a very fast processor is used, we still cannot claim that it is more energy efficient to use the 802.11b radio instead of the 802.15.4. The reason is that the faster processor usually comes at the expense of higher power consumption which adds to the total energy consumption of the sensor node. As a result of this, solely considering the energy consumption of the radio is not enough. Instead, the energy consumption of the processor used to drive the radio should also be considered. *Therefore the question: "Which radio is more energy efficient?", now becomes: "Which pair of processor/radio combination is more energy efficient?"*. To better illustrate this, consider the case where the 802.15.4 radio is interfaced to an MSP430 processor (low bandwidth) and the 802.11b radio is interfaced to an ARM processor (high bandwidth). The ratio of energy consumptions of these two configurations is as follows:

$$\frac{E_{highbandwidth}}{E_{lowbancwidth}} = \frac{E_{ARM} + E_{802.11b}}{E_{MSP430} + E_{802.15.4}} \qquad (5)$$

Note, that the terms E_{MSP430} and E_{ARM} refer to the energy consumption of the processors due to radio I/O operations only and do not include energy spent on other processing tasks.

According to equation (5), in order to find out which configuration is the most energy efficient, it is *not* enough to simply compare the energy consumption of the two radios. In other words, $E_{802.11b} < E_{802.15.4}$ does not imply that $E_{highbandwidth} < E_{lowbandwidth}$ and vice versa . *Thus, we have to jointly consider the energy consumption of the radio and the processor*.

6 The Network Traffic Effect

So far, in our experiments only one node was accessing the radio channel. However, in real sensor network deployments multiple sensor nodes with overlapping transmission ranges have to concurrently transmit data. We show that this need to share the radio channel can impact radio energy consumption in various ways.

In particular, we quantify the impact of sharing the radio channel on packet transmission time and radio energy consumption for different network parameters. We ran our experiments under a typical one-hop neighborhood of a sensor network where a node is exposed to RF traffic from multiple neighboring nodes. We did so by scaling the number of sensor nodes, that were concurrently transmitting, from one up to five. Each sensor node was periodically generating 1000 packets for transmission. Different experiments were run while varying the number of nodes used, the period at which each node generated packets and the size of the transmitted packets. The experiments were run using a group of 802.15.4 radios and a group of 802.11 radios.

Note that the purpose of these experiments was not to characterize the MAC layer of the two radios in absolute terms. Different application parameters, communication patterns, hardware platforms, and operating environments can have a significant impact on such a characterization. Hence, our experiments aimed at revealing major trends and tradeoffs, and providing an indication of the impact that the background network traffic can have on radio energy consumption. The reader should not use the experimental results presented in this section as an absolute characterization of the MAC layer of the two radios.

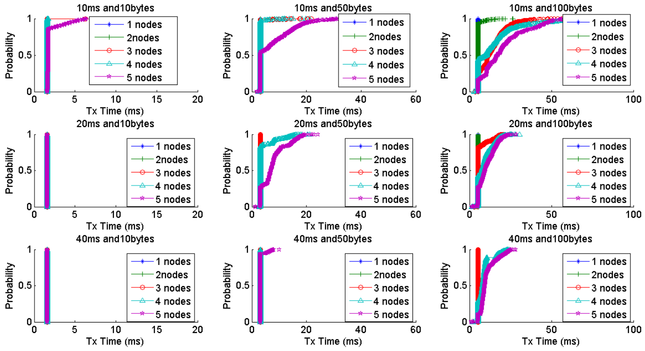

Figure 12. Empirical CDF of a single 802.15.4 packet transmission time for a given number of nodes and across different parameters.

6.1 The 802.15.4 Radio Performance

Figure 12 shows the empirical cumulative distribution of the packet transmission time for 802.15.4 radios as we vary the packet generation period, the packet size, and the number of nodes. Each column of plots shows the effect of increasing the packet generation period for a fixed packet size. Each row of plots shows the effect of increasing the packet size under a fixed packet generation period. The impact due to varying these parameters can be clearly seen in the first row and the last column where the network traffic is high. Initially (10ms and 10 bytes) the effect on packet's transmission time is negligible even with 5 nodes. However, as we increase the size of each packet, the impact due to network traffic becomes noticeable. *In particular, with a 50 byte packet and 5 nodes, the packet transmission time can become more than twice the minimum time with a probability of ∼40%. With 100 byte packets, not only does this probability jump to approximately 70% but the same effect appears with 4 or even 3 nodes.*

This increase in transmission time can be explained as follows. The increased packet size results in longer channel usage for a single sensor node. However, since each node generates packets at a fixed rate, the probability of a packet transmission attempt getting backed off due to detecting a busy radio channel increases as the packet size increases. A backed off packet transmission attempt causes the packet to be buffered for a later transmission attempt, resulting in an

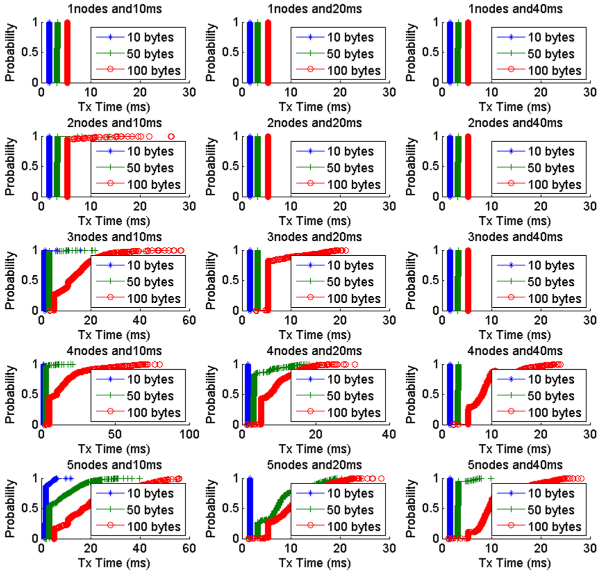

Figure 13. Empirical CDF of a single 802.15.4 packet transmission time for a given packet size and across different parameter values.

increased transmission time. In addition to this extra delay, when a new packet is submitted to the MAC layer and the packet queue is not empty, the new packet has to wait for all the packets in the queue to be transmitted (or dropped) resulting in an increased transmission time.

Similarly, when the packet size is fixed and the packet generation period decreases the packet transmission time increases. For instance, as Figure 12 shows (last column) when packets of 100 bytes are generated every 40ms, the packet transmission time increases only when 5 nodes are used. In contrast, when packets of 100 bytes are generated every 10ms, the packet transmission time is significantly affected when 4 or even 3 nodes are used.

The same trends can also be seen in Figure 13. The offset of the different lines in every plot corresponds to the difference, in terms of transmission time, for different packet sizes. It is clear, that as the number of nodes increases, the packet transmission time also increases, since more nodes have to share the radio channel. However, the effect on the packet transmission time is more intense when larger packet sizes or smaller packet generation periods are used.

Figure 15 shows that the per-packet radio energy consumption follows the same trends as the packet transmission time. As the number of nodes increases, radio channel congestion increases and therefore the radio energy consumption increases. However, we again observe that the packet size has a major impact. With 5 nodes and a packet generation period of 10ms, the average per-packet energy consumption is only slightly affected for a packet payload of 10 bytes. *In contrast, when a payload of 50 or 100 bytes is used, the average per-packet energy consumption can be as much as 2 or even 3 times higher respectively*. Again, as the packet generation period increases the per-packet energy dissipation decreases.

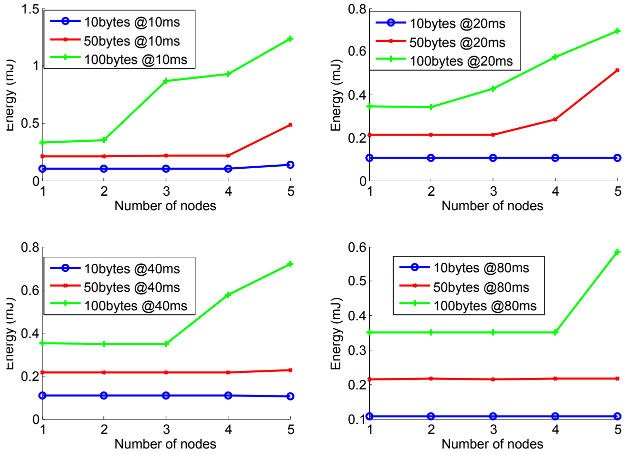

Figure 15. Per-packet 802.15.4 energy dissipation as a function of different network parameters.

6.2 The 802.11b Radio Performance

Measuring the effect of network traffic on the performance of the high-bandwidth 802.11b radio was more challenging than that of the 802.15.4 radio. Instead of capturing the variation in the overall packet transmission time, as defined in Section 4, we decided to measure the variation in the time that elapses from when we start sending the "send packet" command to the radio (the packet has already been structured and written to the radio) until the time the processor has been notified about the successful or unsuccessful transmission of the packet. This measurement methodology was used because the time it takes for the processor to write a packet to the radio is very large due to the complex interface of the 802.11b radio and the limited capabilities of the processor. Given the high data rate of the 802.11b radio (11Mbps) the actual RF transmission time is almost negligible when compared to the time it takes to contruct and write the packet to the radio. Thus, in order to observe transmission time variations we had to ignore the constant overhead of constructing and writing the packet to the radio.

Figure 14 shows the empirical cumulative distribution of the packet transmission time as we vary the packet generation period, the packet size, and the number of nodes. Each column of plots shows the effect of increasing the packet transmission period for a fixed packet size. Each row of plots shows the effect of increasing the packet size for a fixed packet generation period. In contrast to the 802.15.4 radio, in all cases, even when the maximum number of nodes is used along with the maximum packet size and the

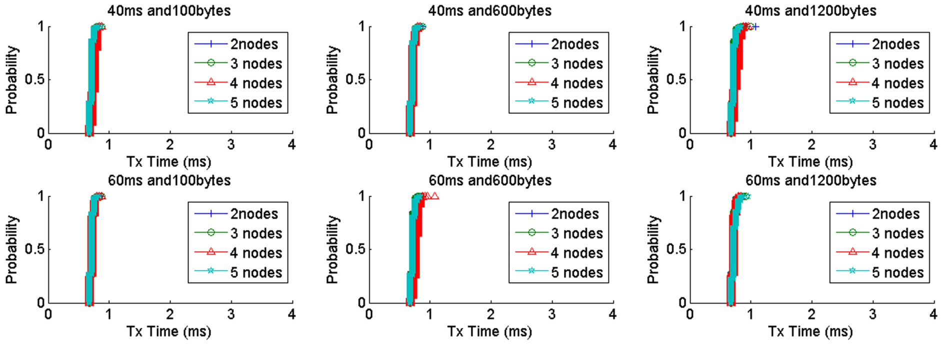

Figure 14. Empirical CDF of a single 802.11b packet transmission time for a given number of nodes and across different parameters.

minimum generating period, there is no significant variation in the packet transmission time for the 802.11b radio. The explanation for this is twofold. First, due to the mismatch between the MSP430 processor and the complex 802.11b radio interface, it is very difficult, if not impossible, to stretch-test the MAC layer of the 802.11b radio. This is because the processor cannot write packets to the radio fast enough to create congestion on the radio channel. For instance, as demonstrated in Section 5, it takes almost 15ms to write a packet of 800 bytes payload to the radio card, while it takes less than 1ms to transmit it on the RF channel. Second, the radio's MAC processor is much faster than the MSP430 microprocessor. This enables the 802.11b radio to quickly back-off and retransmit a packet compared to the time it takes for the low power processor to write a packet to the radio.

6.3 Re-Thinking Energy Efficiency

Quantifying the impact of packet size on the average packet transmission time and energy consumption unveils an important design tradeoff that is often overlooked. Under no bit errors, it is widely accepted that it is more energy efficient to transmit as many bytes as possible in a single packet. This is due to the fact that the overhead of constructing a packet and sending the headers of a packet occurs a smaller number of times, resulting in less energy being wasted with larger packets. Indeed, our measurements in Section 5 verified this for both radios. However, these measurements did not take into account the presence of network traffic on the radio channel which can significantly affect the radio energy consumption. As a result, the answer to the following question becomes non-trivial:

What is the most energy efficient way of transmitting a specific amount of data in a wireless sensor network?

To better demonstrate this, we use the computed average packet energy consumption from Figure 15 to answer this question under different network traffic conditions. In particular, even though different packet sizes were used, we set the overall number of bytes to be sent to be always the same and equal to 100000 bytes. This corresponds to 1000, 2000 and 10000 packets of size 100, 50 and 10 bytes respectively. The overall energy consumption due to packet transmissions for all the different packet sizes and for different network parameters are shown in Figure 16. In all cases when only one or two nodes are used, the most energy efficient way for transmitting the data is to use the maximum allowable packet size. *However, as the network traffic increases (more sensor nodes or smaller packet generation period), we observe that it is more energy efficient, by up to 50% in certain cases, to reduce the packet size in half or reduce the packet generation period.* In these cases, when maximum-size packets are used, the channel becomes congested resulting in increased energy consumption due to multiple transmit attempts and buffering at the MAC layer. As Figure 16 shows, in many cases this overhead can nullify the energy savings due to maximum-size packets. By reducing the packet size, we allow the nodes to better share the radio channel and thus reduce the energy overhead due to MAC buffering and re-transmissions, making the overall communication more energy efficient.

7 Energy Efficient Radio Selection

The experimental study presented in this paper revealed and quantified various bottlenecks and parameters that have a significant impact on radio energy consumption. These parameters can be roughly classified into two categories: the *off-line* and *on-line* parameters. The former category contains those parameters that do not change over time. In other words, we can compute them once and reuse them as necessary. The latter category contains those parameters that change over time and depend on network characteristics such as background traffic and network topology. These parameters need to be periodically updated to reflect the current network status.

The set of off-line parameters that were identified and studied in detail in Section 5 are: *(1)* radio energy consumption due to RF communication, *(2)* radio energy consumption due to I/O operations, *(3)* radio startup energy overhead and we have considered the processor energy consumption

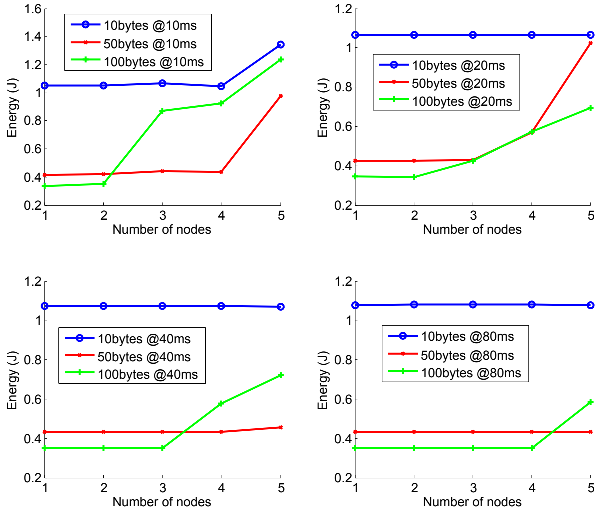

Figure 16. Projected total 802.15.4 energy dissipation for sending 100000 bytes across different parameter values.

for radio I/O operations. Quantifying these parameters using real hardware measurements enables us to build the off-line part of the model that describes a sensor node's communication related energy. This off-line model takes a form similar to Figure 11, which identifies various break-even points between two or more radios given a set of off-line parameters.

Even though such a break-even point is indicative of the relative energy efficiency between two radios, *typically, this alone is not enough* for selecting the most energy efficient radio at a given instance. The reason is that this model ignores the effects due to the run-time behavior of the sensor network, such as the background traffic which can have a significant impact on the energy consumption as described in Section 6.

Such run-time effects are captured by the on-line set of parameters that were partially identified and discussed in Section 6: *(1)* the transmission packet size, and *(2)* the rate the packets are generated, under different background network traffic levels. When the radio channel becomes congested, simplistic energy management techniques that use minimum packet generation period or maximum packet size, can increase rather than decrease the overall energy consumption. For example, Section 6 shows that significant energy savings can be obtained by using a smaller packet size equal to half the maximum size, and by decreasing the packet generation period.

One factor that we have not examined so far is how asymmetry in network topology, due to different transmission ranges of the radios, impacts the radio energy consumption. In general, low power radios like the 802.15.4 have a smaller (a factor of 2 or 3) transmission range compared to high power radios like the 802.11b. Depending on the network topology, the higher transmission range could translate to a smaller break even point for the two radios. For example, a one-hop packet transmission in the 802.11b radio might correspond to a two or three-hop packet transmission in 802.15.4. This effect, when aggregated at the network level, can cause the energy consumption of 802.15.4 to be two or three times higher than that computed from the off-line parameters.

Given the complete set of parameters, both on-line and off-line, one can use the following line of reasoning to determine the most energy efficient radio at a given instance:

(1) Use the off-line parameters to uniquely identify the break-even point between the radios.

(2) After discovering the network topology, take into account the transmission range effect when possible in the calculation of the break-even point.

(3) Under light network traffic use the computed break-even point to choose which radio is the most energy efficient. Under heavy network traffic, properly adjust the packet size and packet generation period at the application level, and recompute the break-even point. Choose the most energy efficient radio based on the re-computed break-even point.

Note that this sequence of steps is only a high level sketch of an algorithm. Developing such an algorithm is beyond the scope of this paper. Instead, this paper focuses on identifying and evaluating the main parameters and design decisions for developing such an algorithm.

8 Discussion

Our experimental findings, presented in the previous sections, can be used as a guideline for designing the next generation energy efficient multi-radio sensor platforms. Our experimental results provide valuable insight and design guidelines on the following three topics:

(1) How can we efficiently interface a heterogeneous set of processors to a heterogeneous set of radios? Our experimental results show that the main problem in interfacing different processors to different radios is the mismatch between the capabilities of the processor and the requirements of the radio interfaces. On the one hand, a low power processor cannot efficiently drive a high-bandwidth radio because the processor is not fast enough, and on the other hand a high-end processor cannot efficiently drive a low-end radio because of the power overhead of the processor. A flexible solution to this problem would be to decouple the different processor and radio interfaces through a bridge. This bridge could be a custom ASIC chip (or a CPLD for experimentation purposes) that on one side implements a simple asynchronous interface for communicating at maximum speed with any type of processor, and on the other side provides efficient high-speed implementations of the different radio interfaces. This enables high-speed commu-

nication between any processor and radio without having to worry about radio specific interface details. This approach is similar to the bus abstraction used in mPlatform [6] for achieving scalable, high-speed communication among heterogeneous processors in a stackable sensor node. A simplified, high-level analysis of this approach, which demonstrated significant energy savings, was first presented in [5].

(2) What is the desired feature set of a high-bandwidth radio specifically designed for sensor network applications? The 802.11b radio was designed for a different set of applications and with a different set of goals in mind compared to wireless sensor networks. The context in which these radios were designed and the intended mechanisms for interfacing them explain the complexity and inefficiency of their I/O management when interfaced to a low-end processor. Ideally, a high-bandwidth radio for sensor networks should combine the simple and efficient I/O and management interface of low power radios such as Chipcon's CC2420 with the high data rates and the embedded MAC layer capabilities of the 802.11 radios. Of course the requirement for higher data rates automatically translates into higher power consumption. However, as long as a low power processor, such as the MSP430 running at a maximum clock speed of 8MHz, can access the radio at an appropriate speed, then it will be able to take full advantage of the energy efficiency of the radio. Current state-of-the-art radios for sensor networks already embed the MAC layer inside the radio chip and offer simple and efficient I/O interfaces [10]. Adding the ability to transmit data at rates similar to the ones of the 802.11b radio would make these radios ideal for the wireless sensor network domain. Another very important factor is the time it takes to configure the radio before it can be used. In general, this cost should be minimized and the whole process should be simple enough, as in the 802.15.4 radio, where less than a few milliseconds are required to power up and start transmitting packets over the radio.

(3) How can we design energy efficient data transmission protocols that are congestion aware? Our experimental results indicate that applying simple data transmission principles, such as always transmitting packets of maximum size, can have a negative impact on radio's energy consumption. Instead, we have identified that by properly adjusting the run-time parameters, such as the packet size and the packet generation period, we can significantly reduce radio's energy consumption under heavy network traffic. It is clear that an energy efficient data transmission protocol has to be aware of the current level of the radio channel congestion, while being able to adjust application level parameters such as the packet size and the packet rate.

References

[1] R. Adler, M. Flanigan, J. Huang, R. Kling, N. Kushalnagar, L. Nachman, C. Y. Wan, and M. Yarvis. Intel mote 2: an advanced platform for demanding sensor network applications. In *SenSys 2005*, pages 298–298, New York, NY, USA, 2005. ACM Press.

[2] Ambicom. WL1100C-CF 802.11b wireless card, http://www.ambicom.com.

[3] M. Buettner, G. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, 2006.

[4] R. Ganti, P. Jayachandran, H. Luo, and T.Abdelzaher. Datalink streaming in wireless sensor networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, 2006.

[5] D. Jung and A. Savvides. An energy efficiency evaluation for sensor nodes with multiple processors, radios and sensors, to appear in infocom 2008.

[6] D. Lymberopoulos, N. Priyantha, and F. Zhao. mPlatform: A reconfigurable architecture and efficient data sharing mechanism for modular sensor nodes. In *IPSN*, 2007.

[7] D. Lymberopoulos and A. Savvides. XYZ: A motion-enabled, power aware sensor node platform for distributed sensor network applications. In *IPSN*, 2005.

[8] D. McIntire, K. Ho, B. Yip, A. Singh, W. Wu, and W. J. Kaiser. The low power energy aware processing (leap)embedded networked sensor system. In *IPSN 2006*, pages 449–457, New York, NY, USA, 2006. ACM Press.

[9] Measurement Computing. USB-2523 DAQ Card, http://www.measurementcomputing.com.

[10] ML7065. Ml7065 - an integrated single chip 802.15.4 radio. http://www2.okisemi.com/site/productscatalog/mcumpu/availabledocs/intro-5434.html.

[11] E. Modiano. An adaptive algorithm for optimizing the packet size used in wireless arq protocols. *Wireless Networks*, 5(4), 1999.

[12] T. Pering, Y. Agarwal, R. Gupta, and R. Want. CoolSpots: Reducing the power consumption of wireless mobile devices with multiple radio interfaces. *MobiSys*, June 2006.

[13] T. Pering, V. Raghunathan, and R. Want. Exploiting radio hierarchies for power-efficient wireless device discovery and connection setup. In *VLSID '05*, pages 774–779, Washington, DC, USA, 2005. IEEE Computer Society.

[14] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *IPSN, SPOTS track*, 2005.

[15] E. Shih, P. Bahl, and M. Sinclair. Wake on wireless: An event driven energy saving strategy for battery operated devices. *Proc. of the Eighth ACM International Conference on Mobile Computing and Networking*, September 2002.

[16] Stargate. Wireless single board computer. http://www.xbow.com/products/xscale.htm.

[17] T. E. Starner. Powerful change part 1: Batteries and possible alternatives for the mobile market. *IEEE Pervasive Computing*, 02(4):86–88, 2003.

[18] T. Stathopoulos, M. Lukac, D. McIntire, J. S. Heidemann, D. Estrin, and W. J. Kaiser. End-to-end routing for dual-radio sensor networks. In *INFOCOM*, 2007.

[19] C.-Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft. Siphon: overload traffic management using multi-radio virtual sinks in sensor networks. In *SenSys '05*, 2005.